# Slate, Carbonate and Research Desktop

1. Apply for Carbonate and Slate Account
2. Install a ThinLinc Client (<https://kb.iu.edu/d/aput>)
3. Login, create a “bin” directory and verify it is in your path.
4. Create a .module configuration file that will load the proper software when you login. <https://kb.iu.edu/d/bcxp> and for more information on modules: <https://kb.iu.edu/d/bcwy>

## Slate / Carbonate

The first step which you will be able to do immediately is to apply for two accounts: “Slate” and “Carbonate”.  Slate is a gigantic file system that houses individual accounts and project accounts which have files that need to be used with the super computers.  Carbonate is a computer cluster (computing nodes, lots and lots of computing nodes) that will be running the processes to analyze data and has a GUI interface.

To view and create accounts go to this address: <https://access.iu.edu/Accounts>

On the left side you’ll see a list of the accounts you already have “created”.  More than likely you do not see Carbonate or Slate in this list unless you created them for a class.  Click on Create Computing Accounts and select Carbonate and click Create Accounts.  You will NOT be storing HIPPA data or any sensitive data so respond no to those questions and yes to any questions about adding references to papers.  Now Slate should be available so Click on Create Computing accounts and this time chose Slate (which is now available after creating the carbonate account).

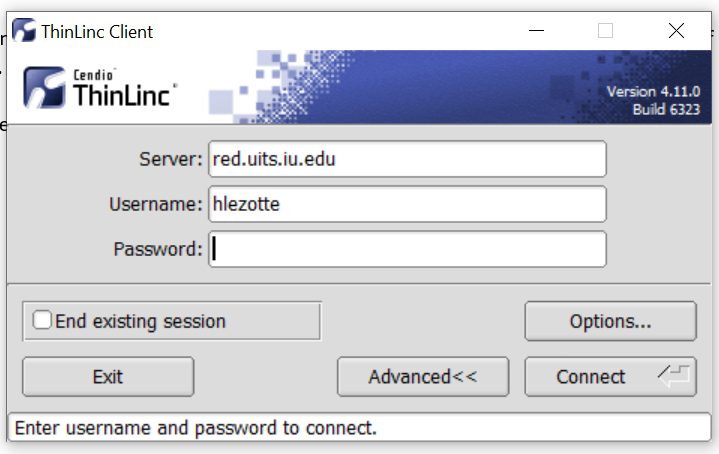
## Install a ThinLinc Client

We will use the ThinLinc Client provided by UITS to access our virtual desktops, currently Redhat 7. You are more than welcome to place this on any machine that you use so that you are able to access your research area from anywhere. There are clients for all of the major operating systems and they are easy to install. Please follow the instruction provided by UITS and pay special attention to the configuration settings on “Screen Size Options”. Those will cause you immense trouble if not followed to the tee. Please read through the KB to make yourself more familiar with RED (Research Desktop). It will help you down the road.

Install instructions: <https://kb.iu.edu/d/aput>

## Login, create a “bin” directory and verify it is in your path

Let’s login. If you followed the ThinLinc instructions then you should have a prompt the looks similar to this:



You will need to use Duo authentication so be on the lookout for a pop-up window that asks you for how you would like to continue to authenticate using Duo.

After you have logged into your desktop, you will be working on the Carbonate Computer Cluster which is part of the Super Computers at IU. Congratulations, you are doing something 99% of the population will never be able to do. Use a Super Computer. Who says science isn’t cool. By the way, at the time of this writing, Big Red 200, IU’s latest super computer is the fastest university owned AI supercomputer in the world. Pretty cool huh? You can read more about Big Red 200 here: https://news.iu.edu/stories/2019/06/iu/releases/07-ai-supercomputer-big-red-200.html

Our first step is to get comfortable with the space. The Research Team has placed the items that you need right off the bat on the desktop. You have a “Log Out” button that will close all of your processes and log you out of the system. You have a “Disconnect ThinLinc Session” that will close the session / window but leave all your processes running. You can use this when you have some long analysis program running and you don’t want to hang out and wait for it to finish. Probably not something you’ll use.

To find out more about the desktop please see: <https://kb.iu.edu/d/apxv>. This KB article is also the default page when you open Firefox on your RED (Research Desktop) so you don’t need to read it now.

*Create a bin directory and check that it is in your path.*

Open terminal by clicking on the icon “Terminal”. By opening up the terminal window you have already run the default settings that has “added” some helpful apps to your terminal settings. We will make some modifications to this in the next step but for now, we move on.

Run the command “pwd” at the terminal prompt to see you default path / user directory. You’ll get something like:

/geode2/home/u030/hlezotte/Carbonate

*\*note: when I say “run the command” in a terminal prompt, I mean type in what I want you to run and hit return. This searches your path (discussed later) and runs the command. I’ll use this often from here on out.*

The first part before your username will change as you are sharing multiple machines with the other super computer users at IU so we’ll only focus on the parts after your username. In the example above, my default user path is “hlezotte/Carbonate”. However, I will need to have the full path in order to move around and since the first part is always changing there is a shortcut naming scheme. My path from the example above is:

/N/u/hlezotte/Carbonate

The “N” covers the folders that precede your username so that you don’t have to worry about which machine you logged into when you start typing path names.

We need a bin directory and it isn’t automatically created

In your default path run the command:

mkdir bin

This will create a bin directory under your default directory and in my case it is:

/N/u/hlezotte/Carbonate/bin

Once this is created let’s double check and make sure you have this folder in your path statement. This is important because it allows us to place executable binaries in the bin folder and run them from any folder in our user account.

To check your path run the command:

echo $PATH

\**note: the capital letters are important. This is case sensitive. Don’t forget the space between “echo” and “$path”.*

What is returned to you is a bunch of paths that are search when you type a command at the terminal prompt. Just like we did above. The very last path entry in the long list should be

/N/u/[your username]/Carbonate/bin

If you don’t have this, stop and get some help from Bob. It’s important that you have this directory available to you in your path statement or the rest of the setup will be for naught. Yes, I used “for naught”, I’m working on my vocabulary but I digress. We are now ready to configure our environment.

## Create a .module file to load the proper software when you login.

Our goal is to have our software environment include anaconda when we login. This isn’t one of the applications loaded by default so we need to modify our environment. Reading this quick kb articles will be very helpful during this section:

* <https://kb.iu.edu/d/bcwy>
* <https://kb.iu.edu/d/bcxp>

To see what software has been loaded by default, at the terminal prompt run the command:

module list

This will return a list of applications that have been loaded automatically and ready to run. There will be a couple of compilers (“gcc” and “intel” in my case), more than like a version of python and a version of perl. There will be others but I want to focus on the version of python.

Anaconda (<https://www.anaconda.com/>) currently comes wrapped in either Python2 or Python3. It doesn’t matter to us which one is available but we’ll use the one with python3 since python 2 is being phased out.

Our problem is it conflicts with the automatically loaded python version. To fix this we need to run two commands.

Command 1 unloads python from our environment. From the terminal prompt run the command:

module unload python

You’ll get something to the affect “Python programing language version 2.7.16 unloaded” returned to you. If python wasn’t loaded, you won’t get anything returned which is good.

To load anaconda we should actually take a look and see what versions are available. To see which versions are available, at the terminal prompt run the command:

module avail anaconda

This will return a list of all the different versions of anaconda that are available to us to load and use. In case you are interested to see ALL the applications available to you to use, at the terminal prompt run the command “module avail”. Any and all of these are available for you to use. Feel free to explore, this is cool stuff.

We want to load anaconda and a version with python 3. Currently we are going to use python 3.7

Command 2 loads anaconda/python3.7. From the terminal prompt run the command:

module load anaconda

You’ll get something returned to you to the affect “anaconda version 2019.03 loaded”. Perfect. If you get an error with anaconda conflictsouring with some other application that is already loaded (python for example), unload the offending application and retry the above command.

*Create a .modules file*

Obviously, we don’t want to do this every time we open a terminal window so we will create a module configuration file that will hold the commands that will unload python and load anaconda.

At the terminal prompt run the command:

nano .modules

*\*note: there is a space after “nano”*

This will place you in a text editor that is editing the new configuration file called “.modules” which the Module system for desktop looks for on terminal startup. This will make our lives much easier.

We need to create two lines in our file. Add the below (don’t copy and paste, type it in without the tab ident):

module unload python

module load anaconda

These are the necessary commands we used above and will run every time we open a terminal window. Save and exit the file (ctrl+o , [return] and ctrl+x *\*note the “carrot” symbol in nano text editor represents the key “ctrl” on the keyboard.*)

Ok, let’s see if it worked. Exit out of the terminal window using the red “x” in the upper right hand corner or run the command from the terminal window “exit”. Reopen the terminal and run the command:

module list

Anaconda should now be one of the applications that you have loaded into your environment. If not, check to make sure that each command is in the .modules file. Run the command:

cat .modules

If the output is anything other than the two command lines you entered earlier, retrace your steps and edit the .modules file again.

## Complete

At this point you now have a Research Desktop environment on Carbonate. There are many help sections about RED in the knowledge base (<https://kb.iu.edu>) and using anaconda is very easy. The base is *Conda* which has some very easy to follow instructions at: <https://docs.conda.io/en/latest/> . Plus, there is always *Google* search which seems to always bring up the needed information.

To use Python3 now in your environment, start the base environment of anaconda (you loaded this earlier) by running the following command at your terminal prompt:

source activate base

This activates the base environment of anaconda and gives you access to now use python3.7 by running:

python

Good Luck!!

Bob